

FAST AND EXACT EVALUATION OF BOX SPLINES VIA THE PP-FORM

JOSHUA HORACSEK AND USMAN ALIM

ABSTRACT. For the class of non-degenerate box splines, we prove that these box splines are piecewise polynomial. This is not a new result, it is in fact a well known and useful property of box splines. However, our proof is constructive, and the main result of this work is a corollary that follows from this proof, namely one that gives an explicit construction scheme for the polynomial pieces in the interior regions of any non-degenerate box spline.

1. INTRODUCTION

Box splines are a multivariate generalization of B-splines [2] that are particularly useful in visualization and signal processing applications. They allow practitioners to tailor approximation schemes (on regular grids) to specific computational and theoretical parameters. These parameters include *support size* (how many samples are needed to reconstruct a value), *smoothness* of reconstruction, and *order of approximation*. There is particular interest in using box splines to span approximation spaces on non-Cartesian lattices, since certain non-Cartesian lattices, like the Body-Centered-Cubic (BCC) lattice, allow more efficient sampling and reconstruction schemes [8].

Typically, to evaluate a box spline, one either relies on the recursive scheme [2, 1, 9], or one must derive an explicit representation of the spline – which has so far been done on a case-by-case basis (see [4] as an example). In this paper, we prove that a box spline’s piecewise polynomial form always exists by explicit construction. We give a set construction recipe that yields the explicit truncated piecewise polynomial form for any box spline. From this we derive the explicit polynomial within each region of evaluation. Once these have been computed, we construct a binary space partitioning tree that splits the support of the box spline into regions of evaluation. Each leaf of the tree then gets an explicit polynomial. To evaluate the box spline, one must simply traverse the tree, then evaluate that polynomial.

When one wishes to evaluate a box spline, the task is not-so trivial. The most widely known evaluation scheme is probably deBoor’s recursive formula [2], which works well in low dimensions, but suffers from numerical issues in higher dimensions [9, 7]. Kobbelt analysed and addressed these issues [1], yet there still appears to be numerical issues in high dimensions [7]. Moreover, these evaluation schemes tend to be fairly slow, since their evaluation schemes are recursive, and rely on solving a system of equations at each step of the recursion.

In the work of Kim et al [7], a fast and stable scheme is derived by creating an indexing system that can quickly identify each region of evaluation. Then within each region of evaluation, the Bernstein Bézier (BB) form of the piecewise polynomial is derived from a stable form of the recursive evaluation formula. However, there are a few pitfalls to their method. Firstly, it relies

Key words and phrases. Multivariate splines, box splines and piecewise polynomials.

on a stable evaluation scheme to derive the BB form of each polynomial. Secondly, its indexing scheme must be derived on a case-by-case basis. Finally, it relies on certain assumptions about the direction vectors that constitute a box spline, namely that they be rational.

The only assumptions we make in this work are that our direction vectors are real valued, and must span the target space. Additionally, the method proposed in this work is exact, our decomposition makes no approximations, and only relies on operations that can be performed exactly within a machine. We do this by incrementally splitting the Fourier representations of a box spline into two parts, a difference operator and a Green's function, then deriving their spatial forms by a series of set constructions. The final result comes from a semi-discrete convolution between these two representations. The paper proceeds as follows. In Section 2 we provide a basic introduction to box splines and the notation we use to proceed with this work. In Section 3, we derive the spatial form for the difference operator via a simple set construction akin (distributionally equivalent) to the inverse discrete Fourier transform of the difference operator. We proceed in Section 4 to construct the spatial form of the Green's function. At first glance, the Fourier form of the Green's function appears to be non-separable, which complicates finding a spatial representation. However, by constructing vectors from the kernel of the matrix that defines a box spline, we can always construct a distributionally equivalent form that is separable, and therefore has an easy to characterize inverse Fourier transform. We then combine those forms via a semi-discrete convolution. In Section 5 we provide our main result, and use it to derive a fast and stable algorithm for exactly evaluating box splines. Finally, in Section 6, we provide example constructions for some known box splines.

2. BACKGROUND

Following the established notation [2], we define a box spline as a generalized function $M_\Xi : \mathbb{R}^s \rightarrow \mathbb{R}$ which is characterized by a list of n column vectors $\xi_1, \xi_2, \dots, \xi_n$, with each $\xi_i \in \mathbb{R}^s$, that are collected in an $s \times n$ matrix $\Xi := [\xi_1 \ \xi_2 \ \dots \ \xi_n]$.

The function M_Ξ is defined recursively by the convolution equation

$$M_{[\]} := \delta \text{ and } M_\Xi := \int_0^1 M_{\Xi \setminus \zeta}(\cdot - t\zeta) dt \quad (1)$$

where δ is the Dirac delta distribution. We say the box spline is non-degenerate if the dimension of the range of Ξ is s . When we speak of the kernel of a matrix, we speak of a specific basis for the null-space of that matrix. We explicitly choose the basis for the kernel so that it is in column echelon form. We use the notation $\ker X$ as shorthand for the null-space of X . For example, $\ker \Xi$ is the $n \times (n - s)$ matrix that forms a basis for the null-space of Ξ and is in column echelon form. We write $z \in \ker X$ if $Xz = 0$.

If a hyperplane defined by columns of Ξ forms an $s - 1$ dimensional subspace of \mathbb{R}^s , then we collect it in the set $\mathbb{H}(\Xi)$. We define the mesh of Ξ as

$$\Gamma(\Xi) := \bigcup_{H \in \mathbb{H}} H + \Xi \mathbb{Z}^n. \quad (2)$$

This is slightly different than the conventional definition of $\Gamma(\Xi)$, projecting \mathbb{Z}^n along Ξ allows us to consider real valued direction vectors in our construction. We write $h \in \Gamma(\Xi)$ to denote a shifted hyper-plane.

We denote the Fourier equivalence between two functions with the symbol “ \longleftrightarrow ”. For example, the function $f(\mathbf{x})$ has the Fourier transform $\hat{f}(\boldsymbol{\omega})$, which we also write as $f(\mathbf{x}) \longleftrightarrow \hat{f}(\boldsymbol{\omega})$. We use the convention that the Fourier transform is defined distributionally as

$$\hat{f}(\boldsymbol{\omega}) := \int f(\mathbf{x}) \exp(-i\boldsymbol{\omega} \cdot \mathbf{x}) d\mathbf{x}.$$

Of pivotal importance to us, is the Fourier representation of a box spline

$$\widehat{M}_{\Xi}(\boldsymbol{\omega}) = \prod_{j=1}^n \frac{1 - \exp(-i\boldsymbol{\omega} \cdot \boldsymbol{\xi}_j)}{i\boldsymbol{\omega} \cdot \boldsymbol{\xi}_j} \quad (3)$$

which we split into two functions, the difference operator

$$\widehat{\nabla}_{\Xi}(\boldsymbol{\omega}) := \prod_{j=1}^n 1 - \exp(-i\boldsymbol{\omega} \cdot \boldsymbol{\xi}_j) \quad (4)$$

and the Green’s function

$$\widehat{G}_{\Xi}(\boldsymbol{\omega}) := \prod_{j=1}^n (i\boldsymbol{\omega} \cdot \boldsymbol{\xi}_j)^{-1} \quad (5)$$

which, when applied with $\widehat{D}_{\Xi} := \prod_{j=1}^n i\boldsymbol{\omega} \cdot \boldsymbol{\xi}_j$ produces the Dirac impulse.

To index an element of a vector $\boldsymbol{\beta}$ we write $\beta(j)$ where j is an integer index. We also define $w(j) := i\boldsymbol{\xi}_j \cdot \boldsymbol{\omega}$, and collect all of those terms in the n -dimensional variable \mathbf{w} .

For the α -power function, we use the definition $\mathbf{x}^{\alpha} := \prod_{j=1}^m x(j)^{\alpha(j)}$ where \mathbf{x} is an m -dimensional variable. It should be understood that by $\mathbf{x}^{-\alpha}$ we explicitly mean $1/\mathbf{x}^{\alpha}$. We also adopt the notations

$$[\mathbf{x}]^{\alpha} := \prod_{j=1}^m \frac{x(j)^{\alpha(j)}}{\alpha(j)!}, \quad (6)$$

$$[\mathbf{x}]_{sgn}^{\alpha} := \prod_{j=1}^m \frac{(x(j))_{sgn}^{\alpha(j)}}{\alpha(j)!}, \quad (7)$$

$$[\mathbf{x}]_{+}^{\alpha} := \prod_{j=1}^m \frac{(x(j))_{+}^{\alpha(j)}}{\alpha(j)!}, \quad (8)$$

for the normalized, normalized signed and normalized one-sided α -power functions respectively. In the above, $(x)_{sgn}^k := x^k \text{sgn}(x)/2$ and $(x)_{+}^k := x^k H(x)$ where $H(x)$ is the heaviside distribution. We write $\|\alpha\|_0$ to count the non-zero elements within α . If α is an n -dimensional vector, we can write the $s \times \|\alpha\|_0$ matrix Ξ_{α} , which denotes the subset of Ξ that are selected by the non-zero $\alpha(j)$. Finally we use $\mathbf{0}, \mathbf{1}$ to denote vectors consisting of all 0 or 1 respectively, and the dimension of the vector should be inferred from context in which it is being used.

The crux of this work is establishing explicit representations of the spatial forms of the difference operator and the Green’s function of the box spline. The final piecewise polynomial form will result from the semi-discrete convolution of these two forms.

3. DIFFERENCE OPERATOR

The difference operator has a simple characterization in the spatial domain which can be constructed with the help of the following definition.

Definition 3.1. For an $s \times n$ matrix Ξ with full rank, define the sequence of multi-sets S_Ξ by the rules

$$S_{[\]} := \{(1, \mathbf{0})\} \text{ and } S_\Xi := \{(-c, \boldsymbol{\xi} + \mathbf{p}) : (c, \mathbf{p}) \in S_{\Xi \setminus \boldsymbol{\xi}}\} \cup S_{\Xi \setminus \boldsymbol{\xi}}. \quad (9)$$

For each tuple in each multi-set of this sequence, the first element of the tuple is a scalar value and the second element is a vector in \mathbb{R}^s .

From this definition we derive the following lemma.

Lemma 3.2. Let Ξ be an $s \times n$ matrix with full rank, then the difference operator ∇_Ξ for M_Ξ is given by the distribution

$$\nabla_\Xi = \sum_{(c, \mathbf{p}) \in S_\Xi} c \delta(\cdot - \mathbf{p}) \quad (10)$$

where δ is the Dirac delta distribution.

Proof. We show this inductively on the number of columns of Ξ . As a base case, assume Ξ contains a single non-zero direction vector, $\boldsymbol{\zeta}$. By definition, we have

$$S_\Xi = \{(1, \mathbf{0}), (-1, \boldsymbol{\zeta})\}.$$

We also have $\widehat{\nabla_\Xi} = 1 - \exp(i\boldsymbol{\zeta} \cdot \boldsymbol{\omega})$ which has the spatial form $\nabla_\Xi = \delta - \delta(\cdot - \boldsymbol{\zeta})$ which is by definition

$$\nabla_\Xi = \sum_{(c, \mathbf{p}) \in S_\Xi} c \delta(\cdot - \mathbf{p}). \quad (11)$$

For the general case, pick some Ξ with $n > 1$ columns, and suppose that for any Ξ' with $i < n$ columns this result holds. Then split Ξ into an s by $(n - 1)$ matrix Ξ' and column vector $\boldsymbol{\zeta}$, and consider the difference operator $\nabla_{\Xi' \cup \boldsymbol{\zeta}}$. From the Fourier expression we have

$$\widehat{\nabla_\Xi} = \widehat{\nabla_{\Xi' \cup \boldsymbol{\zeta}}} = \prod_{\boldsymbol{\xi} \in \Xi' \cup \boldsymbol{\zeta}} (1 - \exp(-i\boldsymbol{\xi} \cdot \boldsymbol{\omega})) = (1 - \exp(-i\boldsymbol{\zeta} \cdot \boldsymbol{\omega})) \widehat{\nabla_{\Xi'}},$$

which is equivalently stated in the spatial domain as

$$\nabla_\Xi = \nabla_{\Xi' \cup \boldsymbol{\zeta}} = \nabla_{\Xi'} - \nabla_{\Xi'}(\cdot - \boldsymbol{\zeta}). \quad (12)$$

The inductive hypothesis then gives

$$\nabla_{\Xi' \cup \boldsymbol{\zeta}} = \sum_{(c, \mathbf{p}) \in S_{\Xi'}} c \delta(\cdot - \mathbf{p}) + \sum_{(c, \mathbf{p}) \in S_{\Xi'}} -c \delta(\cdot - (\mathbf{p} + \boldsymbol{\zeta})) \quad (13)$$

and it follows, by definition of S_Ξ , that

$$\nabla_\Xi = \nabla_{\Xi' \cup \boldsymbol{\zeta}} = \sum_{(c, \mathbf{p}) \in S_\Xi} c \delta(\cdot - \mathbf{p}) \quad (14)$$

which completes the induction. \square

4. GREEN'S FUNCTION

The Green's function of a box spline allows us to characterize the polynomial within each region of the box spline's mesh. To do this we need the spatial form of the Green's function. Going from the Fourier form to the spatial form of the Green's function is a challenge due to the general non-separability of the Fourier form. Here, we do this by pulling specially crafted vectors from $\ker \Xi$ that, when applied to the Fourier Green's function, can always reduce the Green's function to a separable form. We start with the definition of such vectors.

Definition 4.1. Take Ξ to be an $s \times n$ matrix with full rank, and take any α with dimension n and $s < \|\alpha\|_0 \leq n$. Let $\mathbf{b}'_1 \cdots \mathbf{b}'_n$ be the rows of $\ker \Xi$, then construct an $(n - \|\alpha\|_0) \times (n - s)$ matrix C_α where \mathbf{b}'_j is included in this matrix iff $\alpha(j) = 0$. Let \mathbf{t}_α be the first column vector of $\ker C_\alpha$, then define $\nu_\alpha := \ker \Xi \cdot \mathbf{t}_\alpha$. If $\|\alpha\|_0 = n$ then such a vector does not necessarily exist, in this case we take ν_α as the first column vector of $\ker \Xi$.

By construction ν_α has the following property.

Proposition 4.2. For Ξ and α as above, the vector ν_α has the property that if $\alpha(j) = 0$ then $\nu_\alpha(j) = 0$.

Proof. This is immediate if $\|\alpha\|_0 = n$. So suppose that $s < \|\alpha\|_0 < n$, and suppose that $\alpha(j) = 0$ and $\nu_\alpha(j) \neq 0$. But then we have $\mathbf{b}'_j \cdot \mathbf{t}_\alpha \neq 0$ so it must be that $\mathbf{t}_\alpha \notin \ker C_\alpha$. \square

From this definition, we now construct a sequence of sets that will allow us to simplify the Green's function of any non-degenerate box spline into a separable form.

Definition 4.3. Let Ξ be an $s \times n$ matrix with full rank, define the sequence of sets P recursively by the rules

$$P_0 := \{(1, \mathbf{1})\} \quad (15)$$

$$P_k := \left\{ \left(-c \frac{\nu_\alpha(j)}{\nu_\alpha(m_\alpha)}, \mathbf{r}_j(\alpha) \right) : (c, \alpha) \in P_{k-1}, m_\alpha < j \leq n, \nu_\alpha(j) \neq 0 \right\} \quad (16)$$

with $m_\alpha := \min\{i : \nu_\alpha(i) \neq 0\}$ and the vector function \mathbf{r}_j defined as

$$\mathbf{r}_j : \mathbb{Z}^n \rightarrow \mathbb{Z}^n : \alpha(i) \mapsto \begin{cases} \alpha(i) + 1 & , \text{ if } i = m_\alpha \\ \alpha(i) - 1 & , \text{ if } i = j \\ \alpha(i) & , \text{ otherwise} \end{cases} \quad (17)$$

and $k \leq n - s$. Here, each P_i is a set of tuples where the first element of each tuple is a real scalar value, and the second element is a vector in \mathbb{R}^n .

Before we use this set to decompose the Green's function of M_Ξ , we need some supplementary results.

Proposition 4.4. *If $(c, \alpha) \in P_k$, then for all l such that $m_\alpha < l \leq n$, we have*

$$\alpha(l) \in \{0, 1\}. \quad (18)$$

Proof. If $k = 0$, then this is true by definition. Next, for the inductive case, suppose that this is true for all j such that $0 \leq j < k$, and pick any $(c, \alpha) \in P_k$. Since $k > 0$, there exists a $(d, \beta) \in P_{k-1}$ such that $c = -d \frac{\nu_\beta(l)}{\nu_\beta(m_\beta)}$, and $\alpha = r_l(\beta)$, for some l and m_β . If $m_\alpha \geq m_\beta$, then $\beta(i) \in \{0, 1\}$, for all i such that $m_\beta < i \leq n$, but by construction in equation (16), we know $\alpha(i) \in \{0, 1\}$ for all $m_\beta < i \leq n$. But then $\alpha(i) \in \{0, 1\}$ for all $m_\alpha < i \leq n$, which completes the proof. So suppose that $m_\alpha > m_\beta$. Recall by construction, we have $\nu_\alpha(m_\alpha) \neq 0$ and $\nu_\beta(m_\beta) \neq 0$. Combine this fact with the definition $\nu_\alpha := \ker \Xi t_\alpha$, $\nu_\beta := \ker \Xi t_\beta$, and with the fact that $\ker \Xi$ is in column echelon form, we have

$$\min\{i : t_\alpha(i) \neq 0\} < \min\{i : t_\beta(i) \neq 0\}. \quad (19)$$

Since $t_\beta \in \ker C_\beta$ forces certain elements of β to be zero, and t_α forces the same elements of α to be zero (and perhaps more), we know $t_\alpha \in \ker C_\beta$ as well. But then t_β could not have been the first column of $\ker C_\beta$, since there exists a column with a lower-indexed, non zero row (by equation (19)), and $\ker C_\beta$ is in column echelon form.

There is however, one case left to show, since for $k = 1$ no such t_β exists. Thus if $k = 1$, choose any $(c, \alpha) \in P_1$, and let $(d, \beta) \in P_0$ (chosen as above). Again, by the same logic as above, if $m_\alpha \geq m_\beta$, then the proof is completed. So suppose that $m_\alpha < m_\beta$. But since both $\nu_\alpha \in \ker \Xi$ and $\nu_\beta \in \ker \Xi$, $\nu_\alpha(m_\alpha) \neq 0$ and $\nu_\beta(m_\beta) \neq 0$ and $\ker \Xi$ is in row echelon form, this contradicts our choice of ν_α since it could not have been the first column of $\ker \Xi$. □

Proposition 4.5. *If $(d, \beta) \in P_k$ for $0 \leq k < n - s$, and $(c, \alpha) \in P_{k+1}$, with*

$$c = -d \frac{\nu_\beta(j)}{\nu_\beta(m_\beta)}, \text{ and } \alpha = r_j(\beta) \quad (20)$$

for some $m_\beta < j \leq n$, then

$$\|\alpha\|_0 = \|\beta\|_0 - 1. \quad (21)$$

Proof. We show this directly. Choose any $(d, \beta) \in P_k$ and $(c, \alpha) \in P_{k+1}$ as stated in the proposition. Then, by Proposition (4.4), we know that $\beta(i) \in \{0, 1\}$ for all i such that $m_\alpha < i \leq n$, by construction in equation (16) exactly one of these non-zero $\beta(i)$ will be decremented (and $\beta(m_\beta)$ will be incremented) to become α , thus $\|\alpha\|_0 = \|\beta\|_0 - 1$. □

Proposition 4.6. *If $(c, \alpha) \in P_k$, $0 \leq k \leq n$, then Ξ_α has full rank.*

Proof. For P_0 this holds by assumption. Next, assume this holds for all $(d, \beta) \in P_j$ where $0 \leq j < k$. Pick any $(c, \alpha) \in P_k$. There exists some $(d, \beta) \in P_{k-1}$, such that $\alpha = r_l(\beta)$ with $m_\beta < l \leq n$, for some m_β and l . By construction of α in equation (16), we clearly have either $\Xi_\beta = \Xi_\alpha$, or $\Xi_\beta \neq \Xi_\alpha$. If $\Xi_\beta = \Xi_\alpha$, then the proof is complete, so suppose that $\Xi_\beta \neq \Xi_\alpha$. By Proposition 4.5, these differ by exactly one column vector, specifically, equation (16) tells us that if $\xi_l \in \Xi_\beta$, then $\xi_l \notin \Xi_\alpha$. It remains to show that ξ_l can be written as a linear combination of vectors

only in Ξ_β (not including ξ_l itself). But we also know

$$\sum_{i=1}^n \xi_i \nu_\beta(i) = \mathbf{0} \quad (22)$$

since $\nu_\beta \in \ker \Xi$. By Proposition 4.2, we can write this as

$$\sum_{i=1, i \neq l}^n \xi_i \nu_\beta(i) = -\nu_\beta(l) \xi_l. \quad (23)$$

Note that all the non-zero elements of ν_β correspond to column vectors of Ξ_β thus we can write ξ_l as the weighted sum of vectors from Ξ_β . Since Ξ_β had full rank, Ξ_α has full rank, completing the induction. \square

Lemma 4.7. *Let Ξ be an $s \times n$ matrix with full rank, then*

$$\widehat{G}_\Xi(\omega) = \sum_{(c, \alpha) \in P_k} c \omega^{-\alpha} \quad (24)$$

for $0 \leq k \leq n - s$.

Proof. When $k = 0$ the result is true by definition. Suppose the result holds for all j such that for $0 \leq j < k$. By assumption we have

$$\widehat{G}_\Xi(\omega) = \sum_{(c, \alpha) \in P_{k-1}} c \omega^{-\alpha}. \quad (25)$$

Next, since $\nu_\alpha \in \ker \Xi$ we have $\nu_\alpha \cdot \omega = 0$, re-arranged this is

$$\frac{\nu_\alpha(m_\alpha) w(m_\alpha) - \nu_\alpha \cdot \omega}{\nu_\alpha(m_\alpha) w(m_\alpha)} = 1. \quad (26)$$

So we certainly have

$$\widehat{G}_\Xi(\omega) = \sum_{(c, \alpha) \in P_{k-1}} (c \omega^{-\alpha}) \frac{\nu_\alpha(m_\alpha) w(m_\alpha) - \nu_\alpha \cdot \omega}{\nu_\alpha(m_\alpha) w(m_\alpha)} \quad (27)$$

$$= \sum_{(c, \alpha) \in P_{k-1}} \sum_{l=m_\alpha+1}^n -c \frac{\nu_\alpha(l)}{\nu_\alpha(m_\alpha)} \omega^{-r_l(\alpha)} \quad (28)$$

which is, by definition,

$$\sum_{(c, \alpha) \in P_k} c \omega^{-\alpha}, \quad (29)$$

completing the induction. \square

The next definition helps project n -dimensional vectors into appropriate s -dimensional vectors, which will aid in the transformation from Fourier form to the final spatial form.

Definition 4.8. *For any α of dimension n with $\|\alpha\|_0 = s$ define μ_α as the s -dimensional vector obtained by retaining the non-zero elements of α . Define also the corresponding vector σ_α that indexes the non-zero elements of α .*

For example, if $s = 2$ and $\alpha = (0, 2, 0, 1)$, then $\mu_\alpha = (2, 1)$ and $\sigma_\alpha = (2, 4)$. With this, we now define a helpful auxiliary function.

Definition 4.9. For any α with dimension n and $\|\alpha\|_0 = s$, define the function

$$T_\alpha(\omega) := \prod_{j=1}^s \frac{1}{i\omega(j)^{\mu_\alpha(j)}}.$$

Proposition 4.10. For any α with dimension n and $\|\alpha\|_0 = s$ then

$$T_\alpha(\omega) \longleftrightarrow \llbracket \mathbf{x} \rrbracket_{sgn}^{\mu_\alpha - 1}.$$

Proof. Using the Fourier equivalence

$$\frac{1}{(i\omega)^k} \longleftrightarrow \frac{(x)_{sgn}^{k-1}}{(k-1)!}, \quad k \in \mathbb{Z}^+, \quad (30)$$

we write

$$T_\alpha(\omega) = \prod_{j=1}^s \frac{1}{i\omega(j)^{\mu_\alpha(j)}} \longleftrightarrow \prod_{j=1}^s \frac{(x(j))_{sgn}^{\mu_\alpha(j)-1}}{(\mu_\alpha(j)-1)!} = \llbracket \mathbf{x} \rrbracket_{sgn}^{\mu_\alpha - 1}$$

by definition. □

Proposition 4.11. For any $s \times n$ matrix Ξ with full rank, and any α with dimension n and $\|\alpha\|_0 = s$ then

$$\mathbf{w}^{-\alpha} \longleftrightarrow \frac{\llbracket \Xi_\alpha^{-1} \mathbf{x} \rrbracket_{sgn}^{\mu_\alpha - 1}}{|\det \Xi_\alpha|}.$$

Proof. Since $\|\alpha\|_0 = s$ we can write

$$\mathbf{w}^{-\alpha} = \prod_{j=1}^s \frac{1}{w(\sigma_\alpha(j))^{\mu_\alpha(j)}} \quad (31)$$

$$= \prod_{j=1}^s (i\xi_{\sigma_\alpha(j)} \cdot \omega)^{-\mu_\alpha(j)} \quad (32)$$

$$= T_\alpha(\Xi_\alpha^T \omega), \quad (33)$$

and

$$T_\alpha(\Xi_\alpha^T \omega) \longleftrightarrow |\det \Xi_\alpha^{-1}| \llbracket \Xi_\alpha^{-1} \mathbf{x} \rrbracket_{sgn}^{\mu_\alpha - 1}. \quad (34)$$

□

Lemma 4.12. Let Ξ be an $s \times n$ matrix with full rank, then

$$G_\Xi(\mathbf{x}) = \sum_{(c, \alpha) \in P_{n-s}} \frac{c}{|\det \Xi_\alpha|} \llbracket \Xi_\alpha^{-1} \mathbf{x} \rrbracket_{sgn}^{(\mu_\alpha - 1)}. \quad (35)$$

Proof. Starting with

$$\widehat{G}_\Xi(\omega) = \sum_{(c, \alpha) \in P_{n-s}} c \mathbf{w}^{-\alpha}. \quad (36)$$

then using Propositions 4.5 and 4.11 we obtain the form stated in the lemma, the only thing left to check is that this is well defined, i.e. we must verify that $\det \Xi_\alpha \neq 0$, but this is true by Proposition 4.6. \square

Combining all these small results leads us to the main result of this work.

Theorem 4.13. *Let Ξ be an $s \times n$ matrix that corresponds to a non-degenerate box spline, then the box spline is given by*

$$M_\Xi(\mathbf{x}) = \sum_{(b, \mathbf{p}) \in S_\Xi} \sum_{(c, \alpha) \in P_{n-s}} \frac{b \cdot c}{|\det \Xi_\alpha|} \llbracket (\Xi_\alpha^{-1} \mathbf{x}) - \mathbf{p} \rrbracket_+^{(\mu_\alpha - 1)}. \quad (37)$$

Proof. This is a direct consequence of Lemma 3.2 and 4.12, combined with the fact that the difference operator annihilates all polynomials of degree lower than the order of the spline [2, I.32]. \square

5. FAST AND STABLE EVALUATION

While the form in equation (37) is explicit, it is important to note that it is a distributional equality. Thus, point-wise equality does not necessarily hold. However, when the function is evaluated away from the knot planes, the function will be a polynomial locally, and the evaluation will be numerically stable. Issues only arise when one wishes to evaluate (37) for points on the knot planes, which may be undefined.

To overcome this, we use the knot planes to find the regions of evaluation for the box spline. We take the polyhedron defined by the direction vectors of the box spline, then split that support of the spline with the knot planes, this produces a list of all the regions to which we can assign polynomials. The polynomial within each region R can then be obtained as

$$P_R(\mathbf{x}) := \sum_{(b, \mathbf{p}) \in S_\Xi} \sum_{(c, \alpha) \in P_{n-s}} \frac{b \cdot c}{|\det \Xi_\alpha|} \llbracket (\Xi_\alpha^{-1} \mathbf{c}_R) - \mathbf{p} \rrbracket_+^{(\mu_\alpha - 1)} ((\Xi_\alpha^{-1} \mathbf{x}) - \mathbf{p})^{(\mu_\alpha - 1)} \quad (38)$$

where \mathbf{c}_R is the center of the region R . This can be interpreted as selecting all the Green's polynomials that touch R in the convolution (37), giving a polynomial for only that region.

If a point lies on a knot plane, and the box spline is continuous at that point, then we can simply choose one of the surrounding regions and evaluate $P_R(\mathbf{x})$ at that point, since the continuity of the spline tells us that it does not matter which polynomial piece we choose at boundaries. To obtain a fast evaluation scheme, we calculate all regions R and their corresponding $P_R(\mathbf{x})$, then use the knot-planes to define a binary space partitioning (BSP) tree. The leaves of this tree will correspond to the regions R , thus we associate $P_R(\mathbf{x})$ to each leaf. The computation of this tree is outlined in Algorithm (1) and the evaluation at a point is outlined in Algorithm (2). In the following algorithms there are two object classes, *InternalNode()* and *LeafNode()* that inherit from a base *Node()* class. Additionally, whenever we speak of some hyperplane h , we speak of it as the normal equation $h(\mathbf{x}) = \mathbf{n} \cdot \mathbf{x} - d$ for whatever vector \mathbf{n} and d define that plane.

It is important to note that the choice of h in Algorithm 1 will affect the performance of evaluation; h should be chosen so as to balance the resulting binary tree. As a heuristic, we choose h so that it splits the given polyhedron into polyhedra of roughly equivalent hyper-volumes. On average, the average traversal time should be $O(\log(k))$ where k is the total number of regions of the box spline. A variant on this algorithm is implemented in a SageMath [3] worksheet and is available as supplementary material.

Algorithm 1 Precompute box spline pieces

```

1: procedure PRECOMPUTEBOXSPLINE( $\Xi$ )
2:   /* Returns a tuple containing the polyhedron of the support of  $\Xi$ 
3:   as the first element and the root Node object as the second */
4:   Compute  $S_\Xi$  from Definition 3.1
5:   Compute  $P_{n-s}$  from Definition 4.3
6:   Compute knot-planes  $h \in \Gamma(\Xi)$  such that  $h \cap \text{supp}(\Xi)$ . Collect these in  $H$ 
7:   Calculate the polyhedron  $Q$  for  $\text{supp}(\Xi)$ 
8:   return ( $Q$ ,  $\text{Recurse}(H, Q)$ )
9: procedure RECURSE( $H, Q$ )
10:  /* Returns a Node object */
11:  Choose  $h \in H$  that splits  $Q$ 
12:  if no such  $h$  exists then
13:    Use equation 38 to derive a polynomial peice  $R$  at the center of  $Q$ 
14:    return  $\text{LeafNode}(R)$ 
15:  Split  $Q$  with  $h$  into left and right regions  $A, B$ 
16:  /* Here, left means all points  $x$  that satisfy  $h(x) \leq 0$ , right is the opposite */
17:   $\text{node} \leftarrow \text{InternalNode}()$ 
18:   $\text{node}.h \leftarrow h$ 
19:   $\text{node}.left \leftarrow \text{Recurse}(H \setminus \{h\}, A)$ 
20:   $\text{node}.right \leftarrow \text{Recurse}(H \setminus \{h\}, B)$ 
21:  return  $\text{node}$ 

```

Algorithm 2 Evaluate box spline

```

1: procedure EVALBOXSPLINE( $x, Q, \text{root}$ )
2:   if  $x \notin Q$  then
3:     return 0
4:   return  $\text{RecurseEval}(x, \text{root})$ 
5: procedure RECURSEEVAL( $x, \text{node}$ )
6:   if  $\text{node}$  is a  $\text{LeafNode}$  then
7:     return evaluate node's polynomial at  $x$ 
8:    $h \leftarrow \text{node}.h$ 
9:    $L \leftarrow \text{node}.left$ 
10:   $R \leftarrow \text{node}.right$ 
11:  if  $h(x) \leq 0$  then
12:    return  $\text{Recurse}(x, L)$ 
13:  return  $\text{Recurse}(x, R)$ 

```

6. EXAMPLES

Here we enumerate some examples of box splines. When feasible, we enumerate their polynomial regions (see Tables (1-3)), and P and S sets. When we enumerate the sets for these box

splines, we enumerate simplified versions of the sets, combining elements that correspond to like terms in the final summation.

Example 6.1. *Courant Element:* The first example we consider is the well known Courant element which is constructed by taking the the two Cartesian principle lattice directions and adding the diagonal vector (see Figure (1)). This is given by the direction matrix

$$\Xi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \quad (39)$$

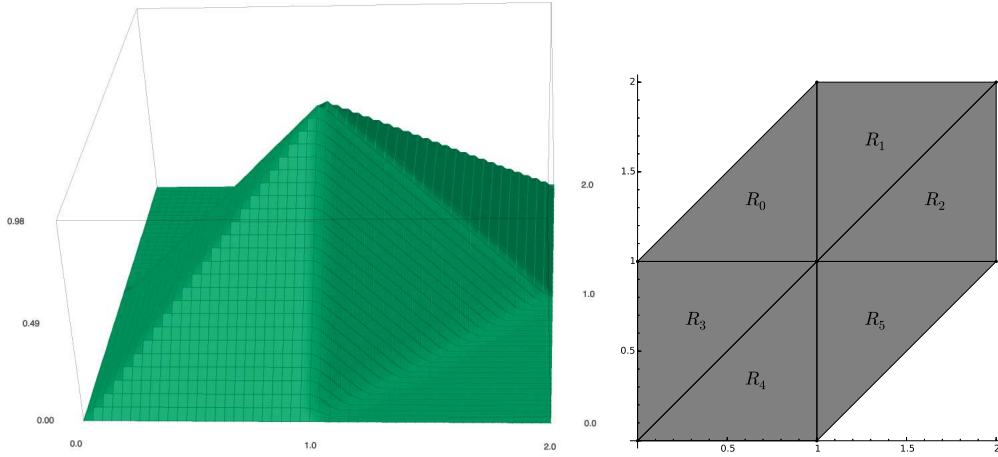


FIGURE 1. On the left is a plot of the Courant element, on the right is a plot of the different polynomial regions of the box spline.

Enumerating the sets for this direction matrix yields

$$\begin{aligned} S_{\Xi} &= \{(-1, (2, 2)), (1, (2, 1)), (1, (1, 2)), (-1, (1, 0)), (-1, (0, 1)), (0, (1, 2))\}, \\ P_1 &= \{(1, (2, 1, 0)), (-1, (2, 0, 1))\}. \end{aligned}$$

Example 6.2. *Zwart-Powell Element:*

The Zwart-Powell element is similar to the Courant element but has an additional vector added along a complementary diagonal. We provide a more thorough example for the construction of P_2 for the ZP element (see Figure (2)). We first start by noting the ZP element's direction matrix and a basis for its null-space

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}, \quad \ker \Xi = \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ -1 & -1 \\ 0 & -1 \end{bmatrix}. \quad (40)$$

We first construct P_1 by definition (4.3). Consider $(c, \alpha) = (1, (1, 1, 1, 1)) \in P_0$. Since this is the base case in definition (4.1), for this α we have

$$\nu_{\alpha} = (1, 1, -1, 0), \text{ and } m_{\alpha} = 1.$$

Definition (4.3) also dictates that this produces elements for all j with $m_\alpha < j \leq n$ such that $\nu_\alpha(j) \neq 0$. For $j = 2$ and $j = 3$ we have the tuples

$$\left(-1\frac{1}{1}, (2, 0, 1, 1)\right) \text{ and } \left(-1\frac{-1}{1}, (2, 1, 0, 1)\right)$$

respectively. Thus we have $P_1 = \{(-1, (2, 0, 1, 1)), (1, (2, 1, 0, 1))\}$, and we apply the recursion to each tuple of this multiset to construct P_2 . For $(c, \alpha) = (-1, (2, 0, 1, 1)) \in P_1$ we construct

$$C_\alpha = \begin{bmatrix} 1 & 2 \end{bmatrix} \text{ which has } \ker C_\alpha = \begin{bmatrix} 2 \\ -1 \end{bmatrix}.$$

Thus $\mathbf{t}_\alpha = (2, -1)$, and we have

$$\nu_\alpha = \ker \Xi \cdot \mathbf{t}_\alpha = (2, 0, -1, 1), \text{ and } m_\alpha = 1,$$

which produces the elements

$$\left(-(-1)\frac{-1}{2}, (3, 0, 0, 1)\right) \text{ and } \left(-1\frac{-1}{2}, (3, 0, 1, 0)\right)$$

for $j = 3$ and $j = 4$ respectively. Finally, we look at the case $(c, \alpha) = (1, (2, 1, 0, 1)) \in P_1$. Again, we construct

$$C_\alpha = \begin{bmatrix} -1 & -1 \end{bmatrix} \text{ which has } \ker C_\alpha = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

This gives $\mathbf{t}_\alpha = (1, -1)$, and we have

$$\nu_\alpha = \ker \Xi \cdot \mathbf{t}_\alpha = (1, -1, 0, 1), \text{ and } m_\alpha = 1,$$

producing the tuples

$$\left(-1\frac{-1}{1}, (3, 0, 0, 1)\right) \text{ and } \left(-1\frac{1}{1}, (3, 1, 0, 0)\right).$$

So we have

$$P_2 = \{(-1/2, (3, 0, 0, 1)), (1/2, (3, 0, 1, 0)), (1, (3, 0, 0, 1)), (-1, (3, 1, 0, 0))\}$$

or, equivalently (by collecting like terms)

$$P_2 = \{(1/2, (3, 0, 0, 1)), (1/2, (3, 0, 1, 0)), (-1, (3, 1, 0, 0))\}.$$

Applying the recursion in definition (3.1) also yields the set

$$\begin{aligned} S_\Xi = & \{(-1, (-1/2, 3/2)), (1, (-1/2, -3/2)), (1, (-3/2, 1/2)), (1, (1/2, 3/2)), \\ & (-1, (1/2, -3/2)), (-1, (-3/2, -1/2)), (1, (3/2, -1/2)), (-1, (3/2, 1/2))\}. \end{aligned}$$

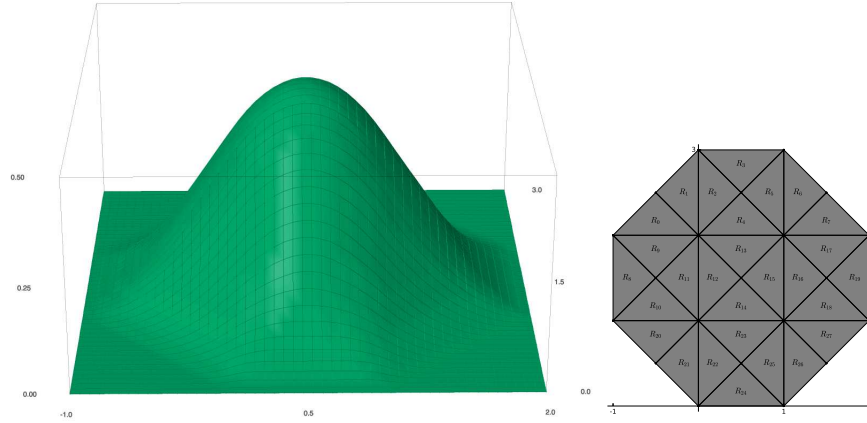


FIGURE 2. On the left is a plot of the ZP-element, on the right is a plot of the different polynomial regions of the box spline.

Example 6.3. *Skewed Element:* As an example of a box spline that has not appeared in the literature, we introduce the four direction skewed element (see Figure (3)) defined by the matrix

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{bmatrix}. \quad (41)$$

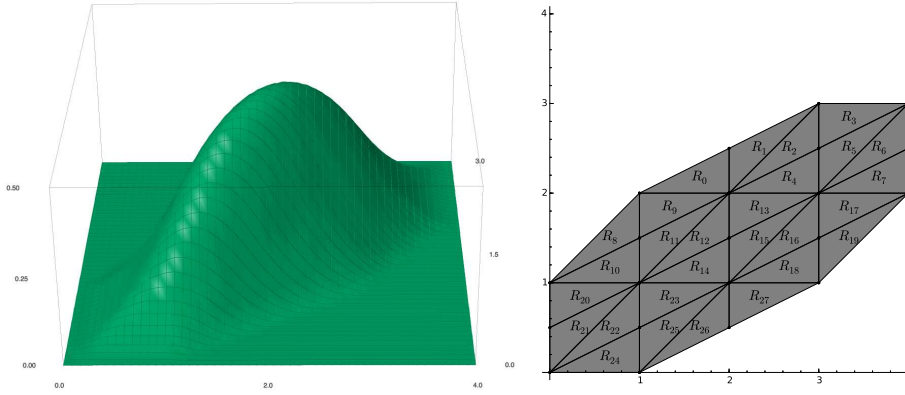


FIGURE 3. On the left is a plot of the Skewed element, on the right is a plot of the different polynomial regions of the box spline.

$$\begin{aligned} S_{\Xi} &= \{(1, (2, 3/2)), (-1, (2, 1/2)), (-1, (-2, -1/2)), (1, (-2, -3/2)), \\ &\quad (1, (-1, 1/2)), (1, (1, -1/2)), (-1, (1, 3/2)), (-1, (-1, -3/2))\} \\ P_2 &= \{(1/2, (3, 0, 0, 1)), (-1, (3, 0, 1, 0)), (-1/2, (3, 1, 0, 0))\}. \end{aligned}$$

Example 6.4. *FCC Cubic Spline:* Finally, as an example of a higher dimensional box spline, we enumerate the sets for the FCC Cubic Spline of Kim et al [6].

$$\Xi = \begin{bmatrix} 1 & 1 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & -1 \end{bmatrix} \quad (42)$$

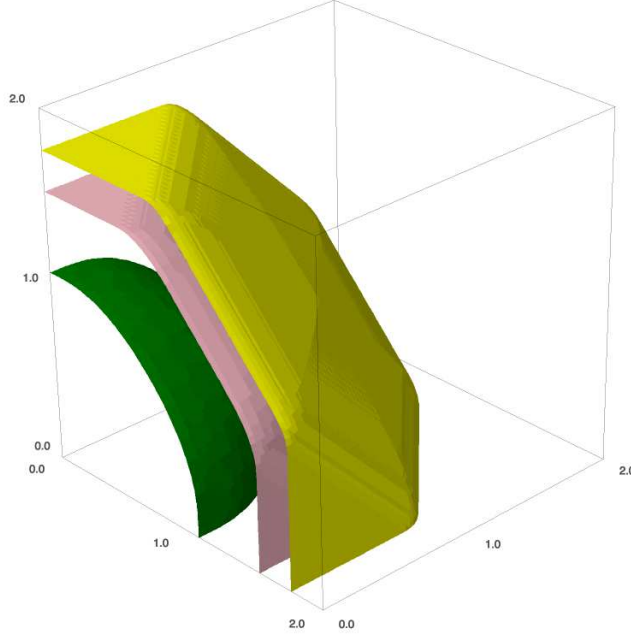


FIGURE 4. Iso-contours for the level sets $1/16$, $1/256$, $1/1024$, pictured in green, pink and yellow respectively.

$$\begin{aligned} S_{\Xi} &= \{(1, (2, 1, 1)), (1, (1, 3, 2)), (-1, (0, 3, 1)), \\ &\quad (1, (-1, 3, 2)), (1, (0, 2, 0)), (-1, (2, 2, 2)), \\ &\quad (1, (0, 0, 4)), (1, (0, 2, 4)), (-1, (0, 3, 3)), \\ &\quad (-1, (-2, 2, 2)), (-1, (-2, 0, 2)), (-1, (2, 0, 2)), \\ &\quad (-1, (0, -1, 1)), (1, (0, 0, 0)), (1, (-2, 1, 3)), \\ &\quad (-1, (0, -1, 3)), (1, (2, 1, 3)), (-1, (-1, 1, 0)), \\ &\quad (1, (-1, -1, 2)), (1, (-2, 1, 1)), (1, (1, -1, 2)), \\ &\quad (-1, (1, 1, 0)), (-1, (-1, 1, 4)), (-1, (1, 1, 4))\} \\ P_3 &= \{(1, (4, 1, 1, 0, 0, 0)), (1, (4, 1, 0, 0, 1, 0)), (-1, (4, 0, 1, 1, 0, 0)), \\ &\quad (1, (3, 2, 1, 0, 0, 0)), (-1, (3, 2, 0, 0, 1, 0)), (1, (4, 0, 0, 0, 1, 1)), \\ &\quad (-1, (4, 1, 0, 0, 0, 1)), (-1, (4, 1, 0, 1, 0, 0)), (1, (3, 2, 0, 0, 0, 1)), \\ &\quad (-1, (3, 2, 0, 1, 0, 0))\}. \end{aligned}$$

As this box spline has many regions, we omit the enumeration of polynomial pieces. Figure (6.4) shows the numerical stability of this form by plotting several iso-contours of this box spline.

7. CONCLUSION

In this work, we provided an explicit decomposition scheme for the piecewise polynomial form of any non-degenerate box spline. While there do exist other evaluation schemes for evaluating box splines, they all typically depend on recursively evaluating a box spline, or they are ad-hoc in their construction procedure. Our method is general and works for all non-degenerate box-splines. However, we only provide a characterization of the splines themselves, when used in an approximation space, it is possible to derive more efficient evaluation schemes for the semi-discrete convolution sum between a box spline and lattice data. This is an avenue for future work.

We also provide, as a SageMath [3, 5] worksheet, the code to derive the piecewise polynomial form of any box-spline, as well as the code to generate BSP trees that represent the evaluation schemes, and code to generate C++ code from those BSPs.

REFERENCES

- [1] de Boor, C.: On the evaluation of box splines. *Numerical Algorithms* **5**(1), 5–23 (1993). DOI 10.1007/BF02109280
- [2] de Boor, C., Höllig, K., Riemenschneider, S.: *Box Splines*. Springer-Verlag New York, Inc., New York, NY, USA (1993)
- [3] Developers, T.S.: Sage Mathematics Software (Version 7.2) (2016). <http://www.sagemath.org>
- [4] Entezari, A., Van De Ville, D., Moller, T.: Practical box splines for reconstruction on the body centered cubic lattice. *Visualization and Computer Graphics, IEEE Transactions on* **14**(2), 313–328 (2008). DOI 10.1109/TVCG.2007.70429
- [5] Horacek, J.: Box-splines. <https://github.com/jjh13/box-splines> (2016)
- [6] Kim, M., Entezari, A., Peters, J.: Box spline reconstruction on the face-centered cubic lattice. *Visualization and Computer Graphics, IEEE Transactions on* **14**(6), 1523–1530 (2008)
- [7] Kim, M., Peters, J.: Fast and stable evaluation of box-splines via the bb-form. *Numerical Algorithms* **50**(4), 381–399 (2009). DOI 10.1007/s11075-008-9231-6
- [8] Kim, M., Peters, J.: Symmetric box-splines on the a^n lattice. *J. Approx. Theory* **162**(9), 1607–1630 (2010). DOI 10.1016/j.jat.2010.04.007
- [9] Kobbelt, L.: Stable evaluation of boxesplines. *Numerical Algorithms* **14**(4), 377–382 (1997). DOI 10.1023/A:1019133501773

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF CALGARY, 2500 UNIVERSITY DR NW, CALGARY, AB T2N 1N4

E-mail address: joshua.horacsek@ucalgary.ca

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF CALGARY, 2500 UNIVERSITY DR NW, CALGARY, AB T2N 1N4

E-mail address: joshua.horacsek@ucalgary.ca

Region	Polynomial Piece
R_0	$x - y + 1$
R_1	$-y + 2$
R_2	$-x + 2$
R_3	x
R_4	y
R_5	$-x + y + 1$

TABLE 1. Polynomial regions of the Courant element.

Region	Polynomial Piece
R_0, R_1	$\frac{1}{4}x^2 - \frac{1}{2}xy + \frac{1}{4}y^2 + \frac{3}{2}x - \frac{3}{2}y + \frac{9}{4}$
R_2	$-\frac{1}{4}x^2 - \frac{1}{2}xy + \frac{1}{4}y^2 + \frac{3}{2}x - \frac{3}{2}y + \frac{9}{4}$
R_3	$\frac{1}{2}y^2 - 3y + \frac{9}{2}$
R_4	$-\frac{1}{2}x^2 + \frac{1}{2}x - \frac{1}{2}y + \frac{5}{4}$
R_5	$-\frac{1}{4}x^2 + \frac{1}{2}xy + \frac{1}{4}y^2 - x - 2y + \frac{7}{2}$
R_6, R_7	$\frac{1}{4}x^2 + \frac{1}{2}xy + \frac{1}{4}y^2 - 2x - 2y + 4$
R_8	$\frac{1}{2}x^2 + x + \frac{1}{2}$
R_9	$\frac{1}{4}x^2 - \frac{1}{2}xy - \frac{1}{4}y^2 + \frac{3}{2}x + \frac{1}{2}y + \frac{1}{4}$
R_{10}	$\frac{1}{4}x^2 + \frac{1}{2}xy - \frac{1}{4}y^2 + y - \frac{1}{2}$
R_{11}	$-\frac{1}{2}y^2 + \frac{1}{2}x + \frac{3}{2}y - \frac{3}{4}$
$R_{12}, R_{13}, R_{14}, R_{15}$	$-\frac{1}{2}x^2 - \frac{1}{2}y^2 + \frac{1}{2}x + \frac{3}{2}y - \frac{3}{4}$
R_{16}	$-\frac{1}{2}y^2 - \frac{1}{2}x + \frac{3}{2}y - \frac{1}{4}$
R_{17}	$\frac{1}{4}x^2 + \frac{1}{2}xy - \frac{1}{4}y^2 - 2x + 2$
R_{18}	$\frac{1}{4}x^2 - \frac{1}{2}xy - \frac{1}{4}y^2 - \frac{1}{2}x + \frac{3}{2}y - \frac{1}{4}$
R_{19}	$\frac{1}{2}x^2 - 2x + 2$
R_{20}, R_{21}	$\frac{1}{4}x^2 + \frac{1}{2}xy + \frac{1}{4}y^2$
R_{22}	$-\frac{1}{4}x^2 + \frac{1}{2}xy + \frac{1}{4}y^2$
R_{23}	$-\frac{1}{2}x^2 + \frac{1}{2}x + \frac{1}{2}y - \frac{1}{4}$
R_{24}	$\frac{1}{2}y^2$
R_{25}	$-\frac{1}{4}x^2 - \frac{1}{2}xy + \frac{1}{4}y^2 + \frac{1}{2}x + \frac{1}{2}y - \frac{1}{4}$
R_{26}, R_{27}	$\frac{1}{4}x^2 - \frac{1}{2}xy + \frac{1}{4}y^2 - \frac{1}{2}x + \frac{1}{2}y + \frac{1}{4}$

TABLE 2. Polynomial regions of the ZP-element.

Region	Polynomial Piece
R_0	$\frac{1}{2}x^2 - xy + \frac{1}{2}y^2 + 2x - 2y + 2$
R_1	$-\frac{1}{2}x^2 + \frac{1}{4}y^2 + 2x - 2y + 2$
R_2, R_3	$\frac{1}{4}y^2 - 2y + 4$
R_4	$-\frac{1}{2}x^2 + xy - \frac{1}{4}y^2 - x - y + \frac{7}{2}$
R_5	$\frac{1}{2}x^2 - 3x + \frac{9}{2}$
R_6	$x^2 - xy + \frac{1}{4}y^2 + x - \frac{1}{2}y + \frac{1}{4}$
R_7	$\frac{1}{2}x^2 - xy + \frac{1}{4}y^2 + 2x - \frac{1}{2}y - \frac{1}{4}$
R_8	$-\frac{1}{2}x^2 + 2x - \frac{1}{2}y - \frac{1}{4}$
R_9, R_{10}	$-x^2 + xy - \frac{1}{2}y^2 + x + \frac{1}{2}y - \frac{3}{4}$
R_{11}	$-\frac{1}{2}x^2 + xy - \frac{1}{2}y^2 - x + \frac{1}{2}y + \frac{5}{4}$
R_{12}	$\frac{1}{2}x^2 - \frac{1}{4}y^2 - 3x + \frac{3}{2}y + \frac{9}{4}$
R_{13}	$x^2 - xy + \frac{1}{4}y^2 - 3x + \frac{3}{2}y + \frac{9}{4}$
R_{14}	$x^2 - xy + \frac{1}{4}y^2 + x - \frac{1}{2}y + \frac{1}{4}$
R_{15}	$\frac{1}{2}x^2 - \frac{1}{4}y^2 + \frac{1}{2}y - \frac{1}{4}$
R_{16}	$-\frac{1}{2}x^2 + xy - \frac{1}{2}y^2 + \frac{1}{2}y - \frac{1}{4}$
R_{17}, R_{18}	$-x^2 + xy - \frac{1}{2}y^2 + x + \frac{1}{2}y - \frac{3}{4}$
R_{19}	$-\frac{1}{2}x^2 + x + \frac{1}{2}y - \frac{3}{4}$
R_{20}	$\frac{1}{2}x^2 - xy + \frac{1}{4}y^2 - x + \frac{3}{2}y + \frac{1}{4}$
R_{21}	$x^2 - xy + \frac{1}{4}y^2 - 3x + \frac{3}{2}y + \frac{9}{4}$
R_{22}	$\frac{1}{2}x^2$
R_{23}	$-\frac{1}{2}x^2 + xy - \frac{1}{4}y^2$
R_{24}, R_{25}	$\frac{1}{4}y^2$
R_{26}	$-\frac{1}{2}x^2 + \frac{1}{4}y^2 + x - \frac{1}{2}$
R_{27}	$\frac{1}{2}x^2 - xy + \frac{1}{2}y^2 - x + y + \frac{1}{2}$

TABLE 3. Polynomial regions of the Skewed element.